Representing Traffic Congestions on Moving Objects Trajectories

Mariano Kohan and Juan M. Ale

Facultad de Ingeniería, Universidad de Buenos Aires
marianokohan@gmail.com, ale@acm.org

Abstract. The discovery of moving objects trajectory patterns representing a high traffic density have been covered on different works using diverse approaches. These models are useful for the areas of transportation planning, traffic monitoring and advertising on public roads. Besides of the important utility, these type of patterns usually do not specify a difference between a high traffic and a traffic congestion. In this work, we propose a model for the discovery of high traffic flow patterns and traffic congestions, represented in the same pattern. Also, as a complement, we present a model that discovers alternative paths to the severe traffic on these patterns. These proposed patterns could help to improve traffic allowing the identification of problems and possible alternatives.

Keywords: Moving objects, trajectories, road network, traffic flow, traffic congestion

1 Introduction

In the last years, there has been a high presence of works related to the data mining of the trajectory data generated by moving objects ([16]). From these works, there has been a lot of attention to the discovery of different type of traffic flow patterns. These patterns can be discovered from trajectories moving inside a road network like [7], [6] and [9] or with a free movement ([5], [11], [4]). Related to the first case, the concept of traffic congestions as a limitation of the road network is considered on more recent papers ([14], [1]). These works are useful on different areas like transport planning, traffic monitoring, carpooling, store locations and advertising on public roads.

In this work, we propose a model for the discovery of high traffic flow patterns in relation to traffic congestions. This relation is displayed according to the shared traffic, like presented in the hot routes model ([7]). Also we present, as a complement, a model that relates these patterns with alternative paths, according to a low traffic level and useful location inside the road network. These models present an increased utility allowing to be applied on additional cases, like the identification of paths inside the road network that require changes (for transport planning), and the redirection of the drivers that contribute to the different congestion areas into alternative paths (for traffic monitoring).

The rest of the paper is organized as follows. Next section comments the works related to this paper. Section 3 describes selected concepts considered from these works. The proposed models are introduced on Section 4. And Sections 5 and 6 gives the definitions and algorithms for each model. Finally, Section 7 concludes this paper with information about the next steps.

2 Related Work

Works about the discovery of traffic flow patterns are related to this paper. The model of hot routes ([7]) is used as the main inspiration for this work because of its balance between the aggregate information about the moving objects and theirs specific behavior (represented in the common traffic in a sequence of edges). Li and others on [7] comments about some alternative methods to discover traffic flow patterns. First, just the aggregate behavior of individuals can be considered connecting only edges in the graph with high traffic. [6] uses this method and complements it with the discovery of the temporal evolution of the patterns. Also, in [9] the model is oriented to the traffic analysis through edges clustering. Another method is to discover moving clusters formed by moving objects, where [5] and [8] are some examples. The third method is about the clustering of trajectories. In this group we can consider some patterns like hot motion paths ([11]), the discovery of the Most Popular Route (MPR) between two locations ([4]) and the distributed parallel clustering method MCR-ACA ([15]). Besides of representing these patterns a high traffic inside the road network, they usually do not consider the cases of traffic jams, where the traffic density is close to the network capacity.

Another group of papers are related to the analysis of traffic congestions. To discover this type of patterns, it is possible to consider the road network characteristics or the moving objects data. In the first case, we have works about representative patterns for the network segments ([2]), usage patterns of road networks ([14], [13]), and the visualization of traffic jams using a GIS map service ([12]). For the second case we can consider diverse patterns like slowly flocks ([10]), the transitions within regions ([17]) and Non-Recurrent Congestion events ([1]). Also, there is the work [3] that considers both types of data in the discovered patterns. The main difference with the current work is that this second group of papers only discover patterns related to traffic congestions, but without relation to patterns with lower levels of traffic.

3 Background Models

The concepts presented in some of the related works are used as a starting point for the models proposed in this paper. In [7], the hot routes patterns are presented. These are traffic flow patterns inside a road network. The road network is represented by a graph $G(V,E)$, where E is the set of edges representing a unit of road segment, and V

the set of vertices representing a street intersection. Also, T is the set of trajectories, with each element composed of an ID (*tid*) and a sequence of edges traveled through: $(tid, <e_1, \ldots, e_k>)$, where $e_i \in E$. The hot routes are built up with a sequence of edges, near each other but not necessarily adjacent, that "...share a high amount of traffic between them." ([7]). The distance between the edges is based on the number of edges inside the road network graph, according to the metric *ForwardNumHops*, which represents the minimum number of edges between the end vertex of two edges. Using this metric, the *Eps-neighborhood* ($N_{Eps}(r)$) of an edge r defines the set of closed edges. The shared traffic considers the same trajectory identification made by each moving object. This model is used as a base for the development of patterns considering traffic congestions.

The work [10] is about the detection of potential traffic jams with slowly flock patterns. In this case, the velocity of each moving object is considered on the discovery of the flock patterns. This idea of using the velocity to identify a traffic jam is applied on the first proposed model.

The discovery of representative patterns for the network segments is proposed on [2]. On this work, the network segments are characterized according to presented network features like length, direction, capacity and density. This last concept of *segment density* $D^*(s)$ allows to identify alternative segments for an edge according to a bounding rectangle (*BR*) covering the segment, and the direction for the edges inside this *BR*. For the second model proposed, this concept is used to discover alternative paths to the traffic congestions.

[6] presents a more general pattern, called dense routes. These patterns are discovered using only the number of objects on each edge of the road network, and adjacent edges are linked if the difference on the number of moving objects is bellow a maximum threshold. A similar idea for the algorithm described on the second model is considered for the discovery of the alternative paths.

4 Models Description

We consider the model of hot routes ([7]) to be most appropriate to discover patterns with heavy traffic in a city road network, because it represents a balance between an aggregate analysis and the behaviors of the individuals. But besides of represent a high density of moving objects in a road network, it does not consider some characteristics of this road network causing the appearance of traffic jams:

- capacity: it is associated with the edges in the road network and represents the maximum number of vehicles that are allowed to circulate into a road segment.
- velocity and time: related to the feature of capacity are the concepts of velocity and time. When the density of objects in a road segment is close to its capacity, the velocity of the moving objects starts to be decreased and the travel time is extended.
- when the velocity and time starts to be affected in an edge, the drivers from the vehicles might choose to continue its path on another alternative segments. This concept of alternative segments is designed as *Density* (D^*) elsewhere.

In this paper, we propose two new models for the discovery of trajectory patterns considering these features:

1. the concept of velocity is considered in order to discover hot routes with jam sections: sections with a density close to its capacity. We call these patterns *jam routes*. This model is described on the next section.
2. the existence of paths that could be used as alternative to the traffic in a *jam route*, because of its location and low density values. These patterns are called *cold routes*. Section 6 presents this model.

5 Discovery of Jam Routes

The *jam route* pattern could be defined as a hot route with one or more subpaths identified as traffic jam. So, it is a path in a road network with heavy traffic (shared by the same objects inside a sliding window) and with one or more sectors having a traffic level close to its capacity.

In order to identify these subpaths the velocity is used. So, each trajectory is composed by its ID (*tid*) and a sequence of pairs representing each edge traveled with its respective mean velocity: $(tid, <(e_1, v_1), (e_2, v_2), \ldots, (e_k, v_k)>)$, where $e_i \in E$ and v_i is the *mean velocity* on e_i.

We consider the use of the velocity to identify traffic congestions is better than compare the density with the road capacity for two reasons:

- the data about the velocity for each moving object on each edge of the trajectory is easier to obtain than the capacity of each edge of the road network
- considering the road segments are part of a network, there also additional factors that could lead to congestions ([14])

The first concept to present is *speed*. It complements the *traffic* definition from [7] to consider the velocity in each edge.

Definition 1 (speed). The *speed(r)* for a given edge *r* is the mean of velocities v_i of the edge *r*.

In order to identify the edges affected by the conditions of a traffic jam the concept *directly traffic jam-reachable* is used.

Definition 2 (directly traffic jam-reachable). An edge *s* is *directly traffic jam-reachable* from an edge *r* with respect to parameters *Eps*, *MinTraffic* and *JamSpeed* if

1. $s \in N_{Eps}(r)$
2. $|traffic(r) \cap traffic(s)| \geq MinTraffic$
3. $speed(s) \leq JamSpeed$ or $speed(r) \leq JamSpeed$

This definition extends the concept of *directly traffic density-reachable* ([7]), to identify traffic jams but maintaining the condition of shared traffic between the edges.

Definition 3 (route traffic jam-reachable). An edge s is *route traffic jam-reachable* from an edge r with respect to parameters *Eps*, *MinTraffic* and *JamSpeed* if

1. *there is a chain of edges* $r_1, r_2, \ldots, r_n$ *with* $r_1=r$ *and* $r_n=s$*, where* r_i *is directly traffic jam-reachable from* r_{i-1} *or* r_i *is just directly traffic density-reachable from* r_{i-1}
2. *for every Eps consecutive edges in the chain,* $|traffic(r_i) \cap traffic(r_{i+1}) \cap \ldots \cap traffic(r_{i+Eps})| \geq MinTraffic$

This definition augments the concept of *route traffic density-reachable* ([7]), allowing to propose a path that relates sections with heavy traffic and sections with traffic jams.

This concept is the base for the discovery of the *jam routes*.

5.1 Algorithm

The algorithm to discover the *jam routes* presents a structure of breadth-first search on the road network graph.

It starts out the discovery from the *hot routes starts* ([7]), verifying if the *speed* on each of these edges is below the *JamSpeed* threshold. In this case, the edge is marked as a *jam*. Next, these *hot routes starts* are extended recursively to form the *jam routes*. The extension is from the last edge, finding the edges inside the N_{Eps} that satisfy the definitions of *directly traffic jam-reachable* or *directly traffic density-reachable*. Then, on each of these possible split edge from the route, the definition of *route traffic jam-reachable* is evaluated (specifically the second condition). If this definition is validated, a new *jam route* is created with the new edge. And, if the added edge is *directly traffic jam-reachable*, it is marked as a *jam*.

The algorithm is called *JamFlowScan* and its pseudo-code is presented as follows:

```
Input: Road network G, object trajectory data T, Eps,
MinTraffic, JamSpeed
Output: Jam routes R

1: Initialize R to {}
2: Let H be the set of hot route starts in G according
to T
3: for every hot route start h in H do
4:   r = new Jam Route initialized to <h> /*mark edge
as "jam" if speed(h) <= JamSpeed */
5:   Add Extend_Jam_Routes(r) to R
6: end for
7: Return R
```

```
Procedure Extend_Jam_Routes(jam route r)
1:  Let p be the last edge in r
2:  Let Q be the set of directly traffic jam-reachable
neighbors of p U the set of directly traffic density-
reachable neighbors of p
3:  if Q is non-empty then
4:    Initialize JR to {}
5:    for every split in Q do
6:      if route traffic jam-reachable condition is
satisfied then
7:        Let r' be a copy of r
8:        Append split's edges to r'
9:        if directly traffic jam-reachable condition
is satisfied then
10:         mark split edge as "jam"
11:       end if
12:       Add Extend_Jam_Routes(r') to JR
13:      end if
14:   end for
15:   return JR
16: else
17:   Return {r}
18: end if
```

To verify the definitions used in the algorithm, the *traffic* set and *speed* for every edge is required. So, the object trajectory data *T* can be converted into table structure that relates each edge with the *tid* of the trajectories that belongs to, and the mean velocity on all the trajectories. The building of this table has linear complexity with respect to the trajectories data.

The jam routes are discovered applying the definitions from the model and identifying the traffic jams on the respective cases:

- initially after the identification of the *hot routes starts* (step 4)
- on the extension of the *jam route* for each split, following the identification of an edge as *route traffic jam-reachable* (steps 9-11 from `Extend_Jam_Routes`).

So, if a traffic congestion is found during the route building, it will be properly identified on the results. Also, the order used to extend the routes adds efficiency to the search but does not omit edges. Therefore, the set of *jam routes* discovered is complete and correct.

6 Discovering Cold Routes

The *cold route* pattern is a path in a road network with low traffic (so it does not affect the network capacity) and with a location inside the road network that allows to be chosen as an alternative path to the traffic present in a *jam route*.

To allow the identification of the alternative routes the concept *BR-neighborhood* $N_{BR}(s)$ is used. It is the same concept *segment density* $D^*(s)$ from [2] (but using a name following the conventions applied to this work): considers the vicinity area of a segment (with a bounding rectangle) and the direction. So, each edge $e \in E$ from the road network graph $G(V,E)$ will be associated with a label representing its direction.

The first concept to present is *cold traffic*. It allows to identify edges with low traffic and that could be considered alternatives to edges with traffic jam (d*irectly traffic jam-reachable*).

Definition 4 (cold traffic). An edge s is considered *cold traffic* with respect to parameters *BR* and *MaxTraffic* if:

1. $|traffic(s)| \leq MaxTraffic$
2. $s \in N_{BR}(s)$ *of directly traffic jam-reachable edge*

Additionally the concept of *directly cold traffic reachable* is presented.

Definition 5 (directly cold traffic reachable). An edge s is considered *directly cold traffic reachable* from an edge r with respect to parameter *MaxTraffic* if:

1. *s is adjacent to r:* $start(s) = end(r)$ *or* $end(s) = start(r)$
2. $|traffic(r)| \leq MaxTraffic$
3. $|traffic(s)| \leq MaxTraffic$

Both concepts are related on the definition of *route cold traffic reachable.*

Definition 6 (route cold traffic reachable). An edge s is considered *route cold traffic reachable* from an edge r with respect to parameters *BR* and *MaxTraffic* if there is a chain of edges $r_1, r_2, \ldots, r_n$ *with* $r_1=r$ *and* $r_n=s$, where:

1. *each* r_i *is directly cold traffic reachable from* r_{i-1}
2. *there exists almost one edge* r_i *that is cold traffic*

The concept of *route cold traffic reachable* allows the discovery of the *cold route* patterns.

6.1 Algorithm

Considering that *cold routes* are formed by edges with low traffic, it is better to discover them using a simple aggregate method.

The proposed algorithm starts the discovery process from the *jam routes* discovered by *JamFlowScan*, finding the *cold traffic* edges according to the N_{BR} of the *directly traffic jam-reachable* edges. Next, these edges are extended to both sides, evaluating the definition *directly cold traffic reachable* into the adjacent edges. With the two conditions of *route cold traffic reachable* satisfied, the new edge is added to the route, considering possible splits (representing different alternative paths).

The algorithm is called *ColdScan* and it pseudo-code is presented as follows:

```
Input: Road network G, object trajectory data T, MaxTraffic, BR, JamRoutes (from JamFlowScan)
Output: Cold routes CR

Initialize CR to {}
Let CS the set of cold traffic edges in G according to T and discovered Jam Routes
for every cold traffic edge cs in CS do
  cr = new Cold Route initialized to <cs>
  Add Extend_Cold_Route_Forward(cs) to CRf
  for every route (extended forward from cs) crf in CRf do
    Add Extend_Cold_Route_Backward(crf) to CR
  end for
end for
return CR

Procedure Extend_Cold_Route_Forward (cold route cr)
Let p be the last edge in cr
Let S be the set of directly cold traffic reachable edges from p with end(p) = start(s)
if S is non-empty then
  Initialize CR to {}
  for every edge s in S
    Let cr' be a copy of cr
    Append edge s to the end of cr'
    Add Extend_Cold_Route_Forward(cr') to CR
  end for
  return CR
else
  return {cr}
end if

Procedure Extend_Cold_Route_Backward (cold route cr)
Let p be the first edge in cr
Let S be the set of directly cold traffic reachable edges from p with start(p) = end(s)
if S is non-empty then
  Initialize CR to {}
  for every edge s in S
    Let cr' be a copy of cr
    Append edge s to the beginning of cr'
    Add Extend_Cold_Route_Backward (cr') to CR
  end for
  return CR
else
  return {cr}
end if
```

The algorithm requires, to verify the definitions used, the *traffic* set for every edge. So, in this case a similar table structure built from the trajectories can be utilized, with a linear complexity with respect to the trajectory data.

ColdScan discovery process applies the definition of *route cold traffic reachable*, considering all the *jam routes* from *JamFlowScan*. Also, these routes are extended to both possible sides. So, the discovered set of *cold routes* is complete.

7 Conclusion and Future Work

In this paper we presented two models for the discovery of traffic flow patterns. The hot routes model for the discovery of high traffic routes is considered as a starting point for the development of patterns representing traffic jams and its alternative paths.

First, in the *jam routes* model the velocity of the moving objects is added in order to identify traffic jam sectors inside the patterns. The relation between these congestions and the high traffic density is according to the shared traffic in common. Next, starting with a vicinity concept the *cold routes* are presented as a path that could be used as an alternative to the traffic in the *jam routes*. These patterns are identified according to a low level of traffic and comparing its location in the road network graph with respect to the congestions in the *jam routes*. The algorithms for the discovery of the proposed models are presented in order to clarify further details.

This is a work in progress. The next step is the implementation of the presented models, in order to compare the discovered patterns with the obtained using some of the related models. This will allow to confirm the utility of these models.

References

1. Berk Anbaroglu, Benjamin Heydecker, and Tao Cheng. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. Transportation Research Part C: Emerging Technologies, 48(0):47 – 65, 2014.
2. Farnoush Banaei-Kashani, Cyrus Shahabi, and Bei Pan. Discovering patterns in traffic sensor data. In: Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS '11, pp. 10–16, New York, USA, 2011. ACM.
3. Kyoung Soo Bok, He Li, Jong Tae Lim, and Jae Soo Yoo. Discovering congested routes using vehicle trajectories in road networks. Advances in Multimedia, 2014.
4. Zaiben Chen, Heng Tao Shen, and Xiaofang Zhou. Discovering popular routes from trajectories. In: Data Engineering (ICDE), 2011 IEEE 27th International Conference on, pp 900–911, 2011.
5. Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. On discovering moving clusters in spatiotemporal data. In: Claudia Bauzer Medeiros, Max J. Egenhofer, and Elisa Bertino, editors, Advances in Spatial and Temporal Databases, volume 3633 of Lecture Notes in Computer Science, pp. 364–381. Springer Berlin Heidelberg, 2005.

6. Ahmed Kharrat, Karine Zeitouni, Iulian Sandu-Popa, and Sami Faiz. Characterizing traffic density and its evolution through moving object trajectories. Fifth International IEEE Conference on Signal-Image Technologies and Internet-Based Systems, pp. 257–263, 2009.
7. Xiaolei Li, Jiawei Han, Jae-Gil Lee, and Hector Gonzalez. Traffic density-based discovery of hot routes in road networks. In: SSTD'07 - Proceedings of the 10th international conference on Advances in spatial and temporal databases, pp. 441–459, Berlin, Heidelberg, 2007. Springer-Verlag.
8. Wenting Liu, Zhijian Wang, and Jun Feng. Continuous clustering of moving objects in spatial networks. In: Ignac Lovrek, Robert J. Howlett, and Lakhmi C. Jain, editors, Knowledge-Based Intelligent Information and Engineering Systems, volume 5178 of Lecture Notes in Computer Science, pp. 543–550. Springer Berlin Heidelberg, 2008.
9. Irene Ntoutsi, Nikos Mitsou, and Gerasimos Marketos. Traffic mining in a road-network: How does the traffic flow? International Journal of Business Intelligence and Data Mining, 3:82–98(17), 2008.
10. Rebecca Ong, Fabio Pinelli, Roberto Trasarti, Mirco Nanni, Chiara Renso, Salvatore Rinzivillo, and Fosca Giannotti. Traffic jams detection using flock mining. In: Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, Machine Learning and Knowledge Discovery in Databases, volume 6913 of Lecture Notes in Computer Science, pp. 650–653. Springer Berlin Heidelberg, 2011.
11. Dimitris Sacharidis, Kostas Patroumpas, Manolis Terrovitis, Verena Kantere, Michalis Potamias, Kyriakos Mouratidis, and Timos Sellis. On-line discovery of hot motion paths. In: Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '08, pp. 392–403, New York, NY, USA, 2008. ACM.
12. Anna Izabel J. Tostes, Fátima de L. P. Duarte-Figueiredo, Renato Assunção, Juliana Salles, and Antonio A. F. Loureiro. From data to knowledge: City-wide traffic flows analysis and prediction using Bing maps. In: Proceedings of the 2Nd ACM SIGKDD International Workshop on Urban Computing, UrbComp '13, pp. 12:1–12:8, New York, NY, USA, 2013. ACM.
13. Junjie Wang, Dong Wei, Kun M. He, Hang Gong, and Pu Wang. Encapsulating urban traffic rhythms into road networks. Scientific Reports, 4, 2014.
14. Pu Wang, Timothy Hunter, Alexandre M. Bayen, Katja Schechtner, and Marta C. Gonzalez. Understanding road usage patterns in urban areas. Scientific Reports, 2, 2012.
15. Jie Yang, Xiaoping Li, Dandan Wang, and Jia Wang. A group mining method for big data on distributed vehicle trajectories in wan. International Journal of Distributed Sensor Networks, 2014.
16. Yu Zheng. Trajectory data mining: An overview. ACM Transaction on Intelligent Systems and Technology, 2015.
17. Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie. Urban computing with taxicabs. In: Ubicomp 2011. ACM, 2011.